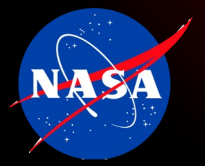
A NASA Perspective on Quantum Computing, with Emphasis on Recent Results in Distributed Computing

Eleanor G. Rieffel

NASA Senior Researcher for Advanced Computing and Data Analytics

NASA Ames Quantum Artificial Intelligence Laboratory (QuAIL) Lead

16 February 2023, IBM Almaden



INTELLIGENT SYSTEMS DIVISION

Why Quantum Computing at NASA?

NASA constantly confronts massively challenging computational problems

Computational capacity limits mission scope and aims

Pleiades supercomputer at NASA Ames

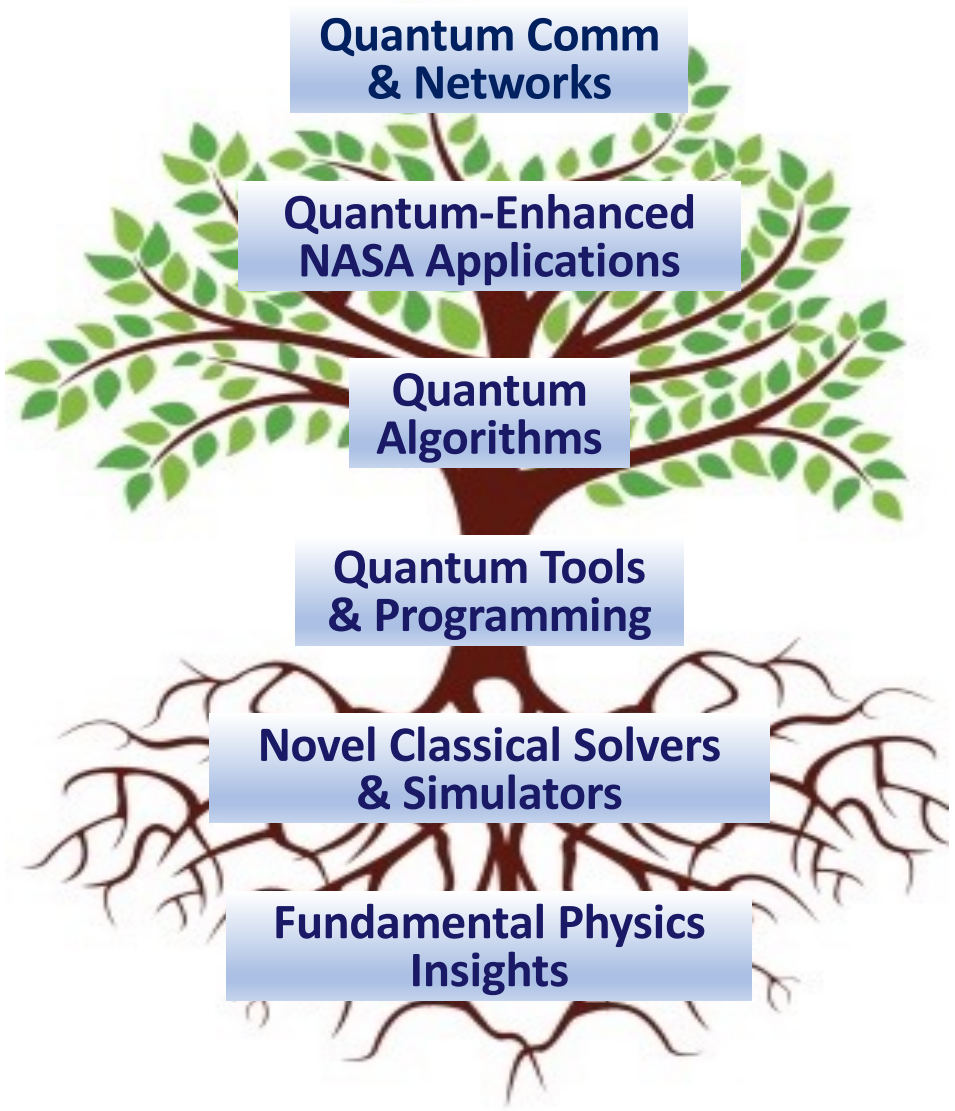
NASA QuAIL mandate: ***Determine the potential for quantum computation to enable more ambitious and safer NASA missions in the future***

Quantum computing has the potential to provide vastly more efficient computation for some applications

- Ability to compute what could not be computed even if every atom in the universe were a classical processor running for the entire age of the universe
  - e.g., factoring, certain material science applications
- Significant speedups in other areas such as optimization, machine learning
- Harnesses uniquely quantum effects

Green computation

- Low energy consumption



Communication & Networks

Quantum networking Distributed QC

Application Focus Areas

Planning and scheduling Material science
Logistics Machine learning

Software Tools & Algorithms

Quantum algorithm design Compiling to hardware
Mapping, parameter setting, error mitigation
Hybrid quantum-classical approaches

Solvers & Simulators

Physics-inspired classical solvers
HPC quantum circuit simulators

Physics Insights

Co-design quantum hardware

New Era for Quantum Computing

INTELLIGENT SYSTEMS DIVISION

Quantum advantage achieved 😀

- Perform computations not possible on even largest supercomputers in reasonable time
- Google – NASA – ORNL collaboration

F. Arute *et al.* (2019), Quantum supremacy using a programmable superconducting processor, *Nature* ***574***, 505-510

… but so far only for a toy problem 😡

- Quantum hardware currently too small for solving practical problems intractable on classical supercomputers
- These devices need to scale up and become more reliable

So what to do in the interim?

- Unprecedented opportunity to invent, explore, and evaluate quantum algorithms *empirically*

At NASA

- Develop quantum and hybrid quantum-classical applications for computational challenges in aeronautics, space, and Earth science
- Algorithms and applications to enable safer, more ambitious, and greater time- and energy-efficient missions

Current status of quantum algorithms

Quantum computing can do everything a classical computer can do

and

Provable quantum advantage known for a few dozen quantum algorithms

Unknown quantum advantage for everything else

Status of classical algorithms

- Provable bounds hard to obtain
  - Analysis is just too difficult
- Best classical algorithm not known for most problems
- Empirical evaluation required
- Ongoing development of classical heuristic approaches
  - Analyzed empirically: ran and see what happens
  - E.g. SAT, planning, machine learning, etc. competitions

A handful of proven limitations on quantum computing

Conjecture: Quantum Heuristics will significantly broaden applications of quantum computing

HybridQ: A Hybrid Quantum Simulator for Large Scale Simulations

INTELLIGENT SYSTEMS DIVISION

Hardware agnostic quantum simulator, designed to simulate large scale quantum circuits

Can run tensor contraction simulations, direct evolution simulation and Clifford+T simulations using the same syntax

Features:

- Fully compatible with Python (3.8+)
- Low-level optimization achieved by using C++ and Just-In-Time (JIT) compilation with JAX and Numba,
- It can run seamlessly on CPU/GPU and TPU, either on single or multiple nodes (MPI) for large scale simulations, using the exact same syntax
- User-friendly interface with an advanced language to describe circuits and gates, including tools to manipulate/simplify circuits.

Recent Improvements:

- Commutations rules are used to simplify circuits (useful for QAOA)
- Expansion of density matrices as superpositions of Pauli strings accepts arbitrary non-Clifford gates,
- Open-source project with continuous-integration, multiple tests and easy installation using either `pip` or `conda`

Open source code available at https://github.com/nasa/HybridQ

S. Mandrà, J. Marshall, E. G. Rieffel, R. Biswas, HybridQ: A Hybrid Simulator for Quantum Circuits, QCS 2021, arXiv:2111.06868

Quantum Distributed Algorithms for Approximate Steiner Trees and Directed Minimum Spanning Trees

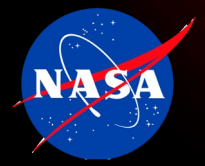
Phillip Kerger, David Bernal Neira, Zoe Gonzales Izquierdo, Eleanor Rieffel

- We provide quantum distributed algorithms to tackle challenging graph problems
  - Approximate Stein Tree Problem
  - Directed Minimum Spanning Tree
- These quantum algorithms provide an asymptotic improvement with respect to the current best known classical algorithm in terms of computational rounds in the CONGEST CLIQUE model
- We provided detailed analysis for the main algorithmic step: finding the all-pairs shortest paths
- We obtained complexity results realizing impractical scales where quantum counterparts become better than classical

NASA

INTELLIGENT SYSTEMS DIVISION

Background

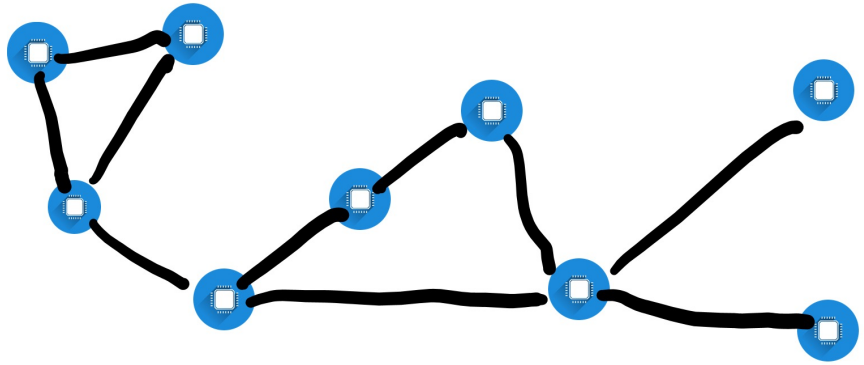


INTELLIGENT SYSTEMS DIVISION

Introduction

Distributed Graph Algorithms: Network of Multiple Processors that communicate

- Model as a graph where the processors are the nodes
- Each node its own starting information, such as list of neighbors



- Example: Network of spacecraft, satellites, or control stations that each have computing power and can communicate
- **Distributed graph algorithms**: Answer some question about that graph, by computing across the processors in a distributed fashion

Introduction: Models

Graph G = (V,E,W) with n = |V| number of nodes, m = |E| number of edges, and W the weights on the edges

CONGEST Model: Aim is to minimize the number of rounds
Computation happens in rounds (compute, communicate, compute, communicate, …)
Congested: Communication limited by message size: each node can send to each of its neighbors O(log(n)) bits each round
— log(n) is length of a node id

Unlimited local computation at each node
Nodes can communicate only with their neighbors

CONGEST-CLIQUE Model:
1., 2., 3. are the same as for Congest Model
4. All nodes can communicate with each other
Key difference: communication graph distinct from graph G

Initial conditions: Each node knows

- its own ID
- the ID's of its neighbors

assuming ID's are 1 to n → log(n) bits to encode

Aim: Answer a question about graph in as few rounds as possible

- Ex: Spanning trees, subgraph detection, shortest paths...

Core Research Question

What problems can benefit from a *quantum* distributed method?

Introduction: Models

Graph G = (V,E,W) with n = |V| number of nodes, m = |E| number of edges, and W the weights on the edges

CONGEST Model: Aim is to minimize the number of rounds
Computation happens in rounds (compute, communicate, compute, communicate, …)
Congested: Communication limited by message size: each node can send to each of its neighbors O(log(n)) bits each round
— log(n) is length of a node id

Unlimited local computation at each node
Nodes can communicate only with their neighbors

CONGEST-CLIQUE Model:
1., 2., 3. are the same as for Congest Model
4. All nodes can communicate with each other
Key difference: communication graph distinct from graph G

Initial conditions: Each node knows

- its own ID
- the ID's of its neighbors

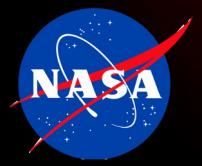
assuming ID's are 1 to n → log(n) bits to encode

Aim: Answer a question about graph in as few rounds as possible

- Ex: Spanning trees, subgraph detection, shortest paths...

Quantum versions: Take CONGEST or CONGEST-CLIQUE, but allow messages to consist of O(log(n)) **qubits**

Prior Results



INTELLIGENT SYSTEMS DIVISION

CONGEST: Negative Results

Main reference: Elkin et al 2012,

“Can Quantum Communication Speed Up Distributed Computation?”

- Proved limitations for quantum CONGEST model
- Quantum communication does NOT provide an improvement for many fundamental problems: **Shortest paths,** Minimum Spanning Tree, **Steiner Tree**, Min Cut, Hamiltonian Cycle…

- **Intuition:** In CONGEST, a significant bottleneck can be communicating between “distant” parts of the network – qubits don’t help with that!

- Elkin, M., Klauck, H., Nanongkai, D., & Pandurangan, G. (2014, July). Can quantum communication speed up distributed computation?. In *Proceedings of the 2014 ACM symposium on Principles of distributed computing* (pp. 166-175).

CONGEST CLIQUE

- Elkin et al.'s results and analysis **do not** carry over to the CONGEST CLIQUE
- So, can quantum communication help in this model?

Surprising **positive results** in in Quantum CONGEST CLIQUE Model (qCCM):

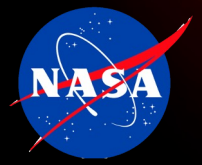
- Faster Triangle Detection, Izumi & Le Gall 2019
- Faster All-Pairs **Shortest-Paths** (APSP), Izumi & Le Gall 2020
  - $\tilde{\mathcal{O}}\left(n^{1/4}\right)$ in quantum versus $\tilde{\mathcal{O}}\left(n^{1/3}\right)$ in classical
  - This was in Elkin's list of problems not admitting speedups!

$$f(n) \in \tilde{\mathcal{O}}\left(g(n)\right) \text{ if } \exists k: f(n) \in \mathcal{O}\left(g(n)\log^k n\right)$$

For which other problems can we exhibit improvements in the quantum CONGEST CLIQUE model?

- Elkin, M., Klauck, H., Nanongkai, D., & Pandurangan, G. (2014, July). Can quantum communication speed up distributed computation?. In *Proceedings of the 2014 ACM symposium on Principles of distributed computing* (pp. 166-175).
- Izumi, T., & Le Gall, F. (2017, July). Triangle finding and listing in CONGEST networks. In Proceedings of the ACM Symposium on Principles of Distributed Computing (pp. 381-389).
- Izumi, T., & Le Gall, F. (2019, July). Quantum distributed algorithm for the All-Pairs Shortest Path problem in the CONGEST-CLIQUE model. In Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (pp. 84-93).

New Algorithms



INTELLIGENT SYSTEMS DIVISION

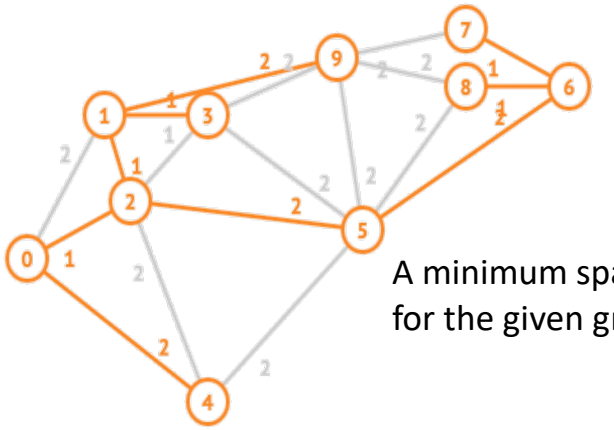
Our Contributions

New algorithms in Quantum CONGEST-CLIQUE Model (qCCM) that succeed with high probability for

- **(approximately optimal) Steiner Trees**
- **Directed Minimum Spanning Trees**

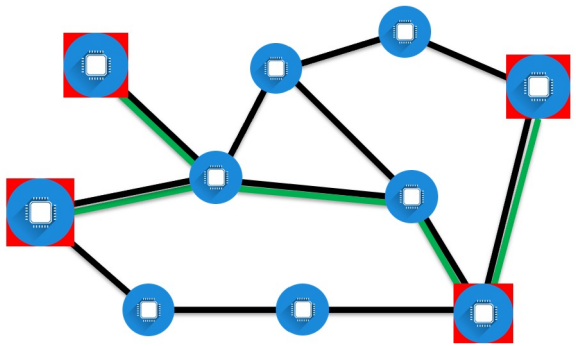
in asymptotically fewer rounds required than for any known classical algorithm

$\rightarrow \tilde{\mathcal{O}}(n^{1/4})$ versus $\tilde{\mathcal{O}}(n^{1/3})$



A minimum spanning tree (orange) for the given graph (grey)

Exact complexity analysis of quantum and classical algorithms reveals improvements needed for both to become practical!

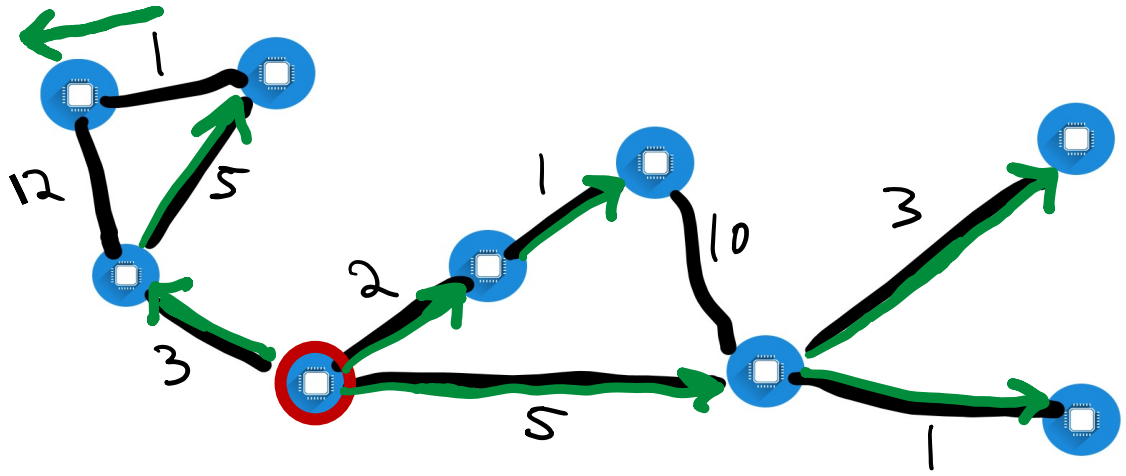


Steiner tree (green) for graph with marked terminal nodes (red)

Contributions: Directed Minimum Spanning Tree

Directed Minimum Spanning Tree:

Given a root node **r** in a directed, weighted graph G, find a directed spanning tree for $\boldsymbol{G}$ rooted at $\boldsymbol{r}$ of minimum weight.



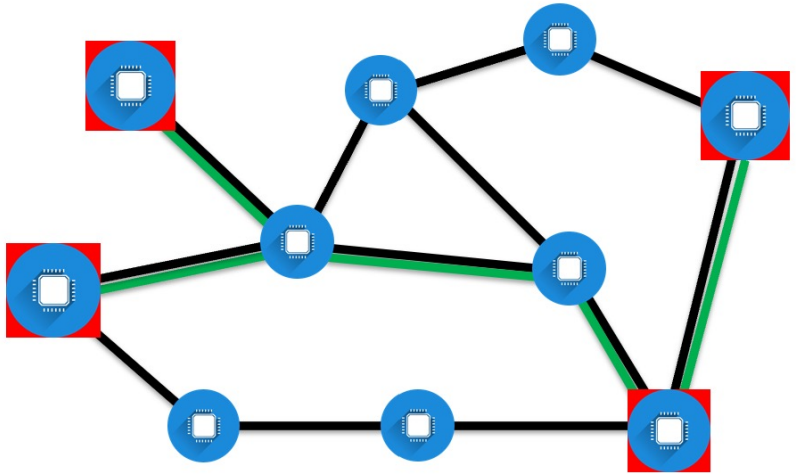
A Directed Minimum Spanning Tree (green) rooted at the red node

Techniques: Make use of fast all-pairs shortest-path with routing tables via triangle finding and distributed Grover Search to obtain the faster algorithms

Contributions: (approx. opt.) Steiner Tree

Steiner Tree:

Given a set of *terminal nodes*, find a minimum weight tree in the graph that contains them



Steiner tree (green) for graph with marked terminal nodes (red)

Techniques: Make use of fast all-pairs shortest-path with routing tables via triangle finding and distributed Grover Search to obtain the faster algorithms

Algorithmic Recipe

Make use of previous techniques such as

1. Distributed Grover Search

2. Triangle Finding

3. Distance Products

4. Shortest Paths and Routing Tables

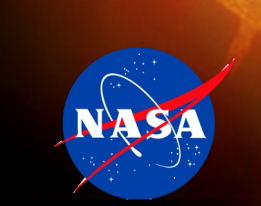
for Steiner Trees and Directed Minimum Spanning Trees problems

Distributed Grover Subroutine

- Suppose a set X of inputs, and a node u, which we call the leader, can evaluate a function $g: X \to \{0,1\}$ in r rounds. Then there exists a quantum distributed algorithm such that u outputs an element $x \in X$ with $g(x) = 1$ using $O\left(r\sqrt{|X|}\right)$ rounds.

- Uses the same ideas as centralized Grover search: search done by leader querying other nodes
- Classical algorithm would use $O(r|X|)$ rounds in absence of additional structure

- Intuition: A node wants to inquire something related to some subset of other nodes – instead of inquiring about one node at a time, use superposition



Aside: Grover's algorithm

INTELLIGENT SYSTEMS DIVISION

Grover's algorithm is the 2nd most famous quantum algorithm after Shor's factoring algorithm

Unstructured search problem: Suppose one has an oracle f: {0, …, N} -> {0,1}. Find a marked element (an x such that f(x) = 1) using as few queries to f as possible

Best possible classically: O(N) queries

Grover's algorithm can find it using only O(√N) queries to a quantum oracle for f that allows superpositions of inputs and outputs a superposition

Intuition: make use of quantum interference effects to get non-solutions to cancel each other

- It is a special case of "quantum amplitude amplification"

It was proved *prior* to Grover's algorithm that quantum algorithms could not do better than O(√N) queries

Triangle Finding

Make use of Distributed Grover Search for finding Triangles!

Idea:

- Triangles live in $V \times V \times V$, so partition that space into n subsets
- Each node searches for triangles in its assigned partition
- Do this via sets of subsets $V_1 \times V_2 \times V_3$, where $|V_1| = |V_2| = n^{1/4}, |V_3| = \mathrm{n}^{1/2}$, and each subset $w \in V_1$ or V_2 contains $n^{3/4}$ nodes, and each $w \in V_3$ contains $n^{1/2}$ nodes.

- So $V_1 \times V_2 \times V_3$ has n elements, that are each a partition of nodes in $V \times V \times V$, each node gets one partition
- Making use of fast routing, the bottleneck becomes searching through the elements of V_3; Grover search can do it in $n^{1/4}$ instead of $n^{1/2}$

Distance Products

Definition 3.2. The *distance product* between two $n \times n$ matrices A and B, also sometimes referred to as the min-plus or tropical product, is defined as:

$$(A \star B)_{ij} = \min_k \{A_{ik} + B_{kj}\}. \tag{3.1}$$

- If W is the adjacency matrix of a graph, then the ij entry of $W \star W$ gives the shortest 2-hop path from node i to node j
- W^n contains shortest n-hop distances , so contains all shortest path distances (exponent w.r.t. $\star$)

Distance Products via Triangles

Proposition 3.3. If $FindTriangleEdges^{-}$ on an n-node integer-weighted graph $G = (V, E, W)$ can be solved in $T(n)$ rounds, then the distance product $A \star B$ of two $n \times n$ matrices A and B can be computed in $T(3n) \cdot \lceil \log_2(\max_{v,z \in G}\{\min_{u \in V}\{A_{vu} + B_{uv}\}\}) \rceil$ rounds.

Proof idea:

- Use negative triangle detection to do a binary search for distances
- Observation: $A_{vu} + B_{uz} < d$ exactly whenever adding an edge of weight $-d$ from z to v gives a negative triangle → do a binary search like this!

Full Steiner Tree Approximation Algorithm in qCCM

Step 1 - APSP and Routing Tables: Solve the APSP problem and add an efficient routing table scheme via triangle finding in $\tilde{O}(n^{1/4})$ rounds

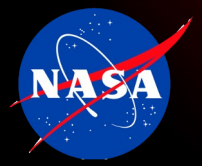
Step 2 - Shortest-path Forest: Construct a shortest-path forest (SPF), where each tree consists of exactly one source terminal and the shortest paths to the vertices whose closest terminal is that source terminal. This step can be completed in one round and n messages

Step 3 - Weight Modifications: Modify the edge weights depending on whether they belong to a tree (set to 0), connect nodes in the same tree (set to ∞), or connect nodes from different trees (set to distance of shortest path between root terminals of the trees that use the edge).

Step 4 - Minimum Spanning Tree: Construct a minimum spanning tree (MST) on the modified graph in $O(1)$ rounds

Step 5 - Pruning: Prune leaves of the MST that are non-terminal nodes, since these are not needed for the Steiner Tree

Complexity

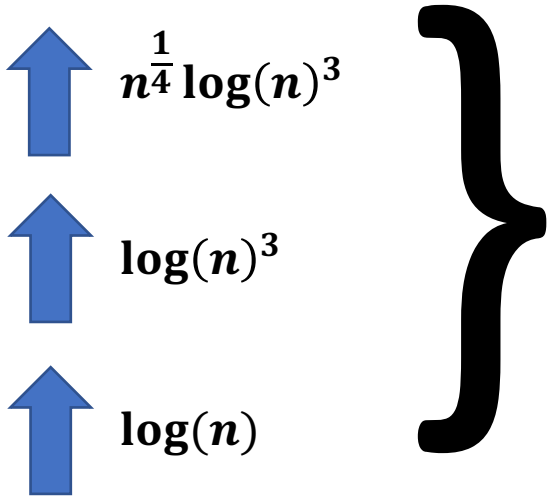


INTELLIGENT SYSTEMS DIVISION

Algorithmic Recipe & Complexity

1. Distributed Grover Search helps with…
2. Triangle Finding helps with…
3. Distance Products helps with…
4. Shortest Paths and Routing Tables helps with…
5. Steiner and Directed Minimum Spanning Trees!

The asymptotic results are exciting!
But more work is needed to bring these algorithms into a practical realm



In CONGEST CLIQUE, can solve anything in n rounds:
To be practical, need

$$3200 \cdot n^{\frac{1}{4}}\log(n)^7 < n$$

for which

$$n > 10^{20}$$

is required!! 10^{11} for classical $\tilde{\mathcal{O}}\left(n^{\frac{1}{3}}\right)$ counterpart.

Future Work

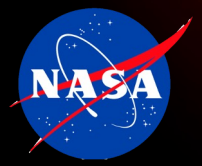
Can improvements be made to have the asymptotic speedup be practical?

Other problems that for which these methods can demonstrate advantages in the quantum setting?

- Make further use of all-pairs shortest paths or fast clique-finding? Speedups through distributed Grover search? Bounded-degree minimum spanning trees?

Other distributed quantum computing models and approaches?

Final Remarks



INTELLIGENT SYSTEMS DIVISION

Take-away points

If we were handed a robust, scalable quantum quantum computer today, for many problems, we would not know what algorithm to run or if quantum computers can help

Lots of work still to be done on quantum algorithms, both

- Heuristic, and
- Those amenable to analysis and proofs

Exciting times ahead, especially as prototype systems improve

Pay attention to quantities hidden in Õ(N) notation!

- Constants and log factors can be important in both the near and long term

Distributed quantum computing is in its infancy, with few results and many open directions

Many opportunities for classical computing to inform quantum computing and to work with or as part of quantum computing

NASA Ames director Hans Mark brought the Illiac IV to NASA Ames in 1972

Illiac IV – first massively parallel computer

- 64 64-bit FPUs and a single CPU
- 50 MFLOP peak, fastest computer at the time

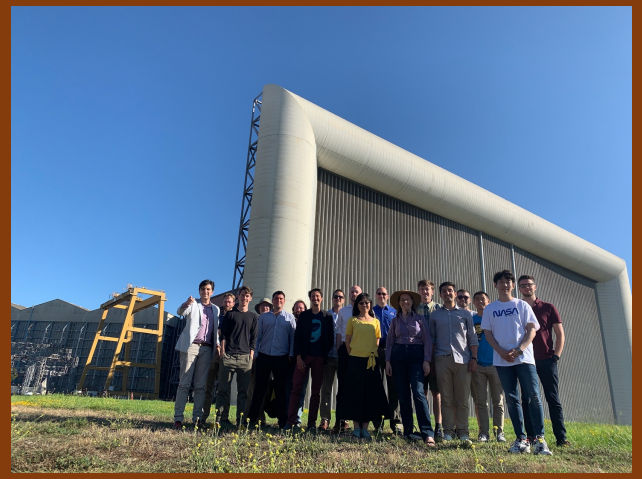
Finding good problems and algorithms was challenging

Questions at the time:

- How broad will the application be of massively parallel computing?
- Will computers ever be able to compete with wind tunnels?

QuAIL Team

With many thanks to everyone on the NASA QuAIL team!



Lead: Eleanor Rieffel, Deputy Lead: Shon Grabbe, Sohaib Alam, Namit Anand, David Bernal Neira, Lucas Brady, Stephen Cotton, Zoe Gonzalez Izquierdo, Stuart Hadfield, Aaron Lott, Salvatore Mandrà, Filip Maciejewski Jeffrey Marshall, Gianni Mossi, Jason Saied, Nischay Suri, Norm Tubman, Davide Venturelli, Zhihui Wang